1	2	3	4	T
A	A	A+	A+	10

Sebastián Taboh 185/13	orden: 13	1 21/02/17.

① Notemos que podemos escribir lo siguiente (justificación del otro lado).

$$f(x,y) = \min_{t \leq cota(x,y)} \left((\forall i)_{1\leq \,\leq |y|} \left(t[i] = \prod_{j=1}^{|x|} p_j^{x[j]\,y[i]} \right) \right)$$

cota(x, y) debe ser p.r y suficientemente grande.

$$cota(x,y) = \prod_{i=1}^{|y|} \left(p_i^{\prod_{j=1}^{|x|} p_j^{\max(x)\cdot\max(y)}} \right)$$

donde max(x) devuelve el máximo de la lista x.

$$\max(x) = \min_{t \leq x} \left((\forall j)_{\leq |x|} \, (x[j] \leq t) \right)$$

- La función max es p.r pues es la minimización acotada sobre un para todo acotado sobre un predicado p.r. Es decir,
  - $x[j] \leq t$ es un predicado p.r pues el observador de la lista es p.r y el $\leq$ también
  - $(\forall j)_{\leq |x|} \, (x[j] \leq t)$ es entonces p.r porque el $\forall$ es acotado ⊛
  - la minimización acotada sobre el predicado p.r es p.r.

⊛ por algo p.r (el observador longitud es p.r).

- cota(x, y) es p.r pues max es p.r y p_j (que denota al primo número j) es p.r y la potencia es p.r. También lo es la productoria que tiene cota p.r.

Así, elevar p_i a esa potencia p.r es p.r y la productoria sobre los i es p.r.

- Al igual que para max y para cota, se puede argumentar que la reescritura de f es p.r (| |, [ ] sobre listas, p_j, potencia, productoria, igualdad, para todo acotado, la función cota y la minimización acotada son p.r).

- f se puede reescribir así porque da la lista que en cada posición i tenga el resultado ~~de elevar desde el primer primo hasta el primo número long. de x)~~
de multiplicar desde el primer primo elevado a la $x[1].y[i]$ hasta el primo número long. de x elevado a la $x[|x|].y[i]$.

Eso es lo que se quiere.

Si se mira la cota, se ve que sirve porque la respuesta seguro va a ser menor.

(La cota eleva a potencias mayores o iguales que las de la respuesta).

(Cuando se dice que el $\forall$ acotado, ~~el $\exists$ a~~ y la minimización acotada son p.r se está teniendo en cuenta que sus predicados lo son. y sus cotas también.).

A

Sebastián Taboh 185/13	Orden: 13	2 21/02/17.

(2)

a) $f(x) = \begin{cases} 1 & \text{si } \exists z \,/\, \phi_z(y) = \phi_x(y)+1 \ \forall y \\ 0 & \text{si no.} \end{cases}$

Es computable pues es la constante 1 dado que el z siempre existe. ✓
z va a ser el número de programa del siguiente programa:

$$Y \leftarrow \phi_x(x_1)$$
$$Y \leftarrow Y+1$$ ✓

- Es claro que si para cierta entrada de x_1 $\phi_x(x_1)$ se cuelga entonces también se cuelga $\phi_z(x_1)$. ✓
(sería mejor la palabra indefine dado que las ϕ son funciones, no programas).
- Si $\phi_x(x_1)$ no se indefine, entonces $\phi_z(x_1)$ tampoco y vale que $\phi_z(x_1) = \phi_x(x_1)+1$. ✓

b) $\exists$ un programa P /

$$\Psi_P^{(2)}(x,y) = \begin{cases} x+y+\#P & \text{si } x \le y \\ \uparrow & \text{si no.} \end{cases}$$

Es verdadero. ✓

Sea $g(z,x,y) = \begin{cases} x+y+z & \text{si } x \le y \\ \uparrow & \text{si no.} \end{cases}$ ✓

(¿por qué?)
g es parc. comp. Por Teorema de la Recursión, $\exists\, e \,/\, \phi_e(x,y) = g(e,x,y)$. ✓

A

Pero si e es el número de un programa E,
$\Phi_e(x,y) = \Psi_E(x,y)$.

Además, $g(e,x,y) = \begin{cases} x+y+e & \text{si } x \le y \\ \uparrow & \text{si no} \end{cases}$

por def. de g.

O sea, dado que $e = \#E$, y e es tal que $\Phi_e(x,y) = g(e,x,y)$,

$$\Psi_E(x,y) = \Phi_e(x,y) = g(e,x,y) = \begin{cases} x+y+\#E & \text{si } x \le y \\ \uparrow & \text{si no} \end{cases} \checkmark$$

O sea, existe un programa que cumple lo que dice el enunciado. ✓

(Consulté si tenía que justificar que g fuera parcial computable y Ariel dijo que no). OK.

Sebastián Taboh 185/13	Orden: 13.	3 21/02/17.

(3)

a) Si $\phi_x(y)\downarrow$ entonces $\phi_x(y) \leq t$ $\forall$ t tal que $STP(y,x,t)$.

Dem.: La idea es sencilla: si en cada paso ~~instrucción~~ a lo sumo se le suma 1 a Y (a lo sumo porque la instrucción puede no ser sumar 1 y a la vez no hay instrucción que sume más que 1), (o puede no involucrar a Y) el valor de salida de $\phi_x(y)$ (asumiendo que la ejecución termina) "sale" en Y, que tiene un valor que a lo sumo es la cantidad de pasos que tomó la ejecución, llamémosla t_0.

Como todo t que cumple $STP(y,x,t)$ es mayor que t_0 (porque t_0 era el mínimo para el cual habría terminado la ejecución, el mínimo t tal que $STP(y,x,t)$) entonces vale (si $\phi_x(y)\downarrow$):

$$\phi_x(y) = \underbrace{Y}_{\text{(la variable)}} \leq t_0 \leq t \;\forall\, t \text{ tal que } STP(y,x,t)$$

(*)

Se puede demostrar que

$$\Big((\exists t)\; STP(y,x,t)\Big) \Rightarrow \left((\forall t)\;\Big(STP(y,x,t) \Rightarrow r\big(SNAP(y,x,t)\big)[1] \leq t\Big)\right)$$

(Alcanza con lo que sigue al $\Rightarrow$).

b) Por a) sabemos que $\phi_x(y)$ es el mínimo valor que puede tomar un t para el cual

$STP(y, x, t)$.

Esto quiere decir que

$$STP(y, x, \phi_x(y) \dot{-} 1) = 0 \quad \forall \langle x, y \rangle.$$

*

Por tanto, ningún par $\langle x, y \rangle$ cumple

$$\phi_x(y)\downarrow \ \wedge \ STP(y, x, \phi_x(y) \dot{-} 1)$$

(es un $\wedge$ con algo falso).

Esto significa que $S = \emptyset$ y la característica es la constante 0 que es computable.
Además, un conjunto es computable sii es c.e. y co-c.e.
Por tanto, también es c.e y co-c.e.

*

Se puede formalizar la demostración de a) con configuración instantánea y haciendo inducción en la cantidad de pasos para mostrar que la variable Y siempre vale $\leq$ que la cantidad de pasos de la ejecución hasta ese momento en que se mira el valor de Y.

*

Sebastián Taboh 185/13.	Orden: 13	4 21/02/17.

Formalicemos lo de ③ a).

Vamos a hacer inducción en la cantidad de pasos t para ver que $\forall\ t\ \ Y \leq t$, o mejor escrito, $\forall \langle t, y\rangle\ r(SNAP(y, x, t))[1] \leq t$.

Sea y una entrada cualquiera. Veamos que vale

$$\forall\ t\ \ \underbrace{r(SNAP(y, x, t))[1]}_{=Y} \leq t.$$

- Caso base: En $t=0$, $Y=0$ y vale $Y \leq t$.

- Paso inductivo: Para el paso t supongo que $Y \leq t$, qvq en el paso $t+1$ vale $Y \leq t+1$.

  - Si i_{t+1} es $V \leftarrow V \dot{-} 1$ entonces:
    * Y queda igual y ya era $\underset{\text{por H.I.}}{\leq t}$, así que seguro es $\leq t+1$
    * a Y se le resta 1 y dado que era $\leq t$ ahora es $\leq t+1$.
  - Si i_{t+1} es $V \leftarrow V+1$ entonces:
    * V no es Y e Y queda igual y ya era $\leq t$, por tanto es $\leq t+1$.
    * V es Y, y como era $\leq t$, al sumarle 1 seguro es $\leq t+1$.
  - Si i_{t+1} es IF $V \neq 0$ GOTO L entonces Y ya era $\leq t$, al no sumarle nada ahora es $\leq t+1$.

Lo que estoy diciendo es que por a) valía

$$\phi_x(y) = \min_t \ STP(y, x, t).$$

Por tanto, si $\phi_x(y) \dot{-} 1$ es menor estricto que $\phi_x(y)$ seguro no vale $STP(y, x, \phi_x(y) \dot{-} 1)$.

Sólo no vale que $\phi_x(y) \dot{-} 1 < \phi_x(y)$ cuando $\phi_x(y) = 0$, pero por la aclaración del ejercicio sé que $STP(y, x, 0) = 0$ y 0 es justamente el valor de $\phi_x(y) \dot{-} 1$ si $\phi_x(y) = 0$.

Así que en cualquier caso

$$STP(y, x, \phi_x(y) \dot{-} 1) = 0 \quad \forall \langle x, y \rangle.$$

Sebastián Taboh 185/13	Orden: 13.	A+	5. 21/02/17.

(4)

$$T = \{ x : \phi_x(y) \uparrow \ \forall y \}.$$

$$\overline{T} = \{ x : \exists y_0 \ \phi_x(y_0) \downarrow \}.$$

$$= \{ x : (\exists \langle y, t \rangle) \ \cancel{\phi_x(y)} \ STP(y, x, t) \}.$$

~~Como STP es p.r, el existencial (escrito en función de un núme~~

$$= \{ x : (\exists z) \ STP(l(z), x, r(z)) \}.$$

Como l y r son p.r. y STP también, el predicado es computable. Así, el existencial no acotado es parcial computable. (visto en clase del 7/2).

Si definimos $g(x) = (\exists z) \ STP(l(z), x, r(z))$, g es parcial computable y vale

$$\overline{T} = \text{dom } g.$$

Así, $\overline{T}$ es c.e. (T es co-c.e).

Veamos que T no es computable (y por tanto tampoco c.e porque si lo fuera, al ser co-c.e, sería computable).

Supongo que T es computable. Sea f la característica de T.

$$f(x) = \begin{cases} 1 & \text{si } x \in T \\ 0 & \text{si no.} \end{cases}$$

$$f(x) = \begin{cases} 1 & \text{si } \forall y \ \phi_x(y)\uparrow \\ 0 & \text{si no} \end{cases}$$

es computable.

Defino $g(x,y) = \begin{cases} 0 & \text{si } f(x)=1 \\ \uparrow & \text{si no} \end{cases}$

g es parc. comp. pues f es computable.
Entonces por Teo. de la Recursión, $\exists e$ /
$\phi_e(y) = g(e,y)$.

Así,

$f(e)=1 \iff \forall y \ \phi_e(y)\uparrow$ por def. de f.

$\iff \forall y \ g(e,y)\uparrow$ pues el e es / $\phi_e(y) = g(e,y) \ \forall y$.

$\iff \neg(f(e)=1)$ por def. de g.

Llegamos a un absurdo que provino de suponer que f era computable. Así, T no es computable. (pues f no lo es).